96768300

CONTROL DATA CORPORATION

CDC® CYBER 18 PROCESSOR WITH MOS MEMORY

MACRO LEVEL

SYSTEM DESCRIPTION
FUNCTIONAL DESCRIPTION
OPERATING PROCEDURE
PROCESSOR INSTRUCTION DESCRIPTION
INTERRUPT SYSTEM
PROGRAM PROTECT
I/O DEVICES

HARDWARE REFERENCE MANUAL

REVISION RECORD

REVISION	DESCRIPTION
A	Manual released.
(4/77)	
B	Manual revised; includes ECO 14967, which corrects storage reference instructions.
(9/77)	
C	Manual revised to include ECO 21683, adding reference instructions and instruction summaries.
(7/80)	
D	Manual revised; includes ECO DS022118, providing an FCC non-tested label to the preface.
(7/81)	

Publication No.
96768300

REVISION LETTERS I, O, Q, S, X, AND Z ARE NOT USED

© 1977, 1980, 1981
by Control Data Corporation

Printed in the United States of America

Address comments concerning this manual to:

Control Data Corporation
Publications and Graphics Division
4455 Eastgate Mall
La Jolla, California 92037

or use Comment Sheet in the back of this manual.

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual, are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

PAGE	REV
Cover	--
Title Page	--
ii	D
iii/iv	D
v/vi	D
vii	A
viii	A
1-1 thru 1-7	A
2-1 thru 2-6	A
3-1 thru 3-6	A
4-1 thru 4-13	A
4-14 thru 4-16	C
4-16.1/4-16.2	C
4-17 thru 4-19	A
4-20	C
4-20.1/4-20.2	C
4-21	C
4-22	A
4-23	C
4-24	C
4-24.1/4-24.2	C
4-25 thru 4-29	A
5-1	A
5-2	A
6-1	A
6-2	A
7-1 thru 7-4	A
A-1	A
A-2	A
B-1	A
B-2	A
B-3	C
B-4	C
C-1 thru C-8	A
Comment Sheet	D
Cover	--

PREFACE

The micro-programmable processor emulates the 1700 family of computers. Readers of this document should be familiar with the Control Data® 1700 Series computers and their associated hardware. The processor is upward-compatible and has an enhanced instruction capacility.

Additional information on CDC software applicable to the micro-programmable processor system can be found in the following publications:

Publication	Publication Number
1700 Computer System Codes	60163500
Mass Storage Operating System (MSOS) Version 5 Reference Manual	96769400
MS FORTRAN Version 3A/B Reference Manual	60362000
CYBER 18 Micro-Programmable Computer Family Micro Processor Hardware Reference Manual	88973400
CYBER Cross System Version 1 (under SCOPE) Micro Assembler Reference Manual	88988800
CYBER Cross System Version 1 (under SCOPE) Macro Assembler Reference Manual	88988900
CYBER Cross System Version 1 (under NOS and NOS/BE) Micro Assembler Reference Manual	96836400
CYBER Cross System Version 1 (under NOS and NOS/BE) Macro Assembler Reference Manual	96836500
TIMESHARE Version 3 Reference Manual	96768000
Operational Diagnostic System (ODS) Reference Manual	39452100

This equipment generates, uses, and can radiate radio frequency energy, and if not installed and used in accordance with the instructions manual, may cause interference with radio communications. As temporarily permitted by regulation, it has not been tested for compliance with the limits for Class A computing devices pursuant to Subpart J of Part 15 of FCC rules, which are designed to provide reasonable protection against such interference. Operation of the equipment in a residential area is likely to cause interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

TABLE 4-8. ENHANCED STORAGE REFERENCE INSTRUCTION ADDRESSES (Contd)

Addressing Modes	Delta	r	i	Ra	Effective Address (EA)	Address of Next Instruction
Absolute Constant	Δ = 0	0	0	0	P + 2	P + 3
		0	0	1	P + 2 + (1)	P + 3
		0	0	2	P + 2 + (2)	P + 3
		0	0	3	P + 2 + (3)	P + 3
		0	0	4	P + 2 + (4)	P + 3
		0	0	5	P + 2 + (Q)	P + 3
		0	0	6	P + 2 + (A)	P + 3
		0	0	7	P + 2 + (I)	P + 3
16-Bit Storage	Δ = 0	0	1	0	(P + 2)	P + 3
		0	1	1	(P + 2) + (1)	P + 3
		0	1	2	(P + 2) + (2)	P + 3
		0	1	3	(P + 2) + (3)	P + 3
		0	1	4	(P + 2) + (4)	P + 3
		0	1	5	(P + 2) + (Q)	P + 3
		0	1	6	(P + 2) + (A)	P + 3
		0	1	7	(P + 2) + (I)	P + 3
16-Bit Relative	Δ = 0	1	0	0	P + 2 + (P + 2)	P + 3
		1	0	1	P + 2 + (P + 2) + (1)	P + 3
		1	0	2	P + 2 + (P + 2) + (2)	P + 3
		1	0	3	P + 2 + (P + 2) + (3)	P + 3
		1	0	4	P + 2 + (P + 2) + (4)	P + 3
		1	0	5	P + 2 + (P + 2) + (Q)	P + 3
		1	0	6	P + 2 + (P + 2) + (A)	P + 3
		1	0	7	P + 2 + (P + 2) + (I)	P + 3
16-Bit Relative Indirect	Δ = 0	1	1	0	(P + 2 + (P + 2))	P + 3
		1	1	1	(P + 2 + (P + 2)) + (1)	P + 3
		1	1	2	(P + 2 + (P + 2)) + (2)	P + 3
		1	1	3	(P + 2 + (P + 2)) + (3)	P + 3
		1	1	4	(P + 2 + (P + 2)) + (4)	P + 3
		1	1	5	(P + 2 + (P + 2)) + (Q)	P + 3
		1	1	6	(P + 2 + (P + 2)) + (A)	P + 3
		1	1	7	(P + 2 + (P + 2)) + (I)	P + 3

Note: ( ) denotes contents of expression

TABLE 4-9. ENHANCED STORAGE REFERENCE INSTRUCTIONS

Instruction	Mnemonic	Description
Subroutine/Jump Exit F4 = 5 F5 = 0 Rb = 0	SJE	Replace the contents of P with the effective address. This instruction can be used as a jump or subroutine exit. For example, if △= 1 and Ra has been set up by a previous subroutine jump (see below), control will be returned following that subroutine jump. Note that subroutine jumps save the address of the instruction, rather than the next instruction, so the subroutine jump exit may be a two-word instruction (△ = 0) rather than three. For example, the following program makes a subroutine jump at location 1000. Register A will contain 1002 upon entry to the subroutine SUB. Upon completion, SUB will exit to location 1003. 1000 0446 SJA+SUB 1001 5000 1002 2000 1003 . . . . . . 2000 SUB . . . . 2020 0430 SJE- 1, A 2021 5001 CAUTION Since Rb=0, a selection of absolute (r = 0) no indirect (i = 0), and no index register (Ra = 0) will result in an EIN instruction.
Subroutine Jump F4 = 5 F5 = 0 Rb = 1, 2, 3, 4, 5, 6, or 7 r = 1, 2, 3, 4, Q, A, or I	SJr	Load register r with the address of the last word of this instruction (i.e., P + 1 for △≠ 0; P + 2 for △=0). The contents of P are then replaced with the effective address.
Add Register F4 = 8 F5 = 0 Rb = 1, 2, 3, 4, 5, 6, or 7 r = 1, 2, 3, 4, Q, A, or I	ARr	Add (using ones complement arithmetic) the contents of the storage location specified by the effective address to the contents of register r. Operation on overflow is the same as for the ADD instruction. The contents of storage are not altered.

TABLE 4-9. ENHANCED STORAGE REFERENCE INSTRUCTIONS (Contd)

Instruction	Mnemonic	Description
Add and Replace Memory F4 = 8 F5 = 1 Rb→r	RMr	Adds (using ones complement arithmetic) the contents of the r register and the contents of the storage location specified by the effective address. The result replaces the contents of the specified storage location. The original contents of the storage location (specified by the effective address) replace the contents of the A register. The contents of the r register are not altered unless r is the A register. Memory is locked until completion of the instruction. Operation on overflow is the same as for the ADD instruction.
Subtract Register F4 = 9 F5 = 0 Rb = 1, 2, 3, 4, 5, 6, or 7 r = 1, 2, 3, 4, Q, A, or I	SBr	Subtract (using ones complement arithmetic), the contents of the storage location specified by the effective address from the contents of register r. Operation on overflow is the same as for the ADD instruction. The contents of storage are not altered.
Subtract Memory F4 = 9 F5 = 1 Rb→r	SMr	Subtracts (using ones complement arithmetic) the contents of the r register from the contents of the storage location specified by the effective address. The result replaces the contents of the specified storage location. The original contents of the storage location (specified by the effective address) replace the contents of the A register. The contents of the r register are not altered unless r is the A register. Memory is locked until completion of the instruction. Operation on overflow is the same as for the ADD instruction.
AND Register F4 = A F5 = 0 Rb = 1, 2, 3, 4, 5, 6, or 7 r = 1, 2, 3, 4, Q, A, or I	ANr	Form the logical product (AND), bit-by-bit, of the contents of the storage location specified by the effective address and the contents of register r. The result replaces the contents of register r. The contents of storage are not altered.
AND Memory F4 = A F5 = 1 Rb = 1, 2, 3, 4, 5, 6, or 7 r = 1, 2, 3, 4, Q, A, or I	AMr	Form the logical product (AND), bit-by-bit, of the contents of the storage location specified by the effective address and the contents of register r. The result replaces the contents of the storage location specified by the effective address. The original contents of the storage location (specified by the effective address) replace the contents of the A register. The contents of register r are not altered unless r is the A register. Memory is locked until completion of the instruction. This instruction is useful for communication between processors via memory.
EOR Register F4 = B F5 = 0 Rb→r	ERr	Forms the logical difference (exclusive OR), bit-by-bit, of the contents of the storage location specified by the effective address and the contents of the r register. The result replaces the contents of the r register. The contents of storage are not altered.
EOR Memory F4 = B F5 = 1 Rb→r	EMr	Forms the logical difference (exclusive OR), bit-by-bit, of the contents of the r register and the storage location specified by the effective address. The result replaces the contents of the specified storage location. The original contents of the storage location (specified by the effective address) replace the contents of the A register. The contents of the r register are not altered unless r is the A register. Memory is locked until completion of the instruction.

TABLE 4-9. ENHANCED STORAGE REFERENCE INSTRUCTIONS (Contd)

Instruction	Mnemonic	Description
Load Effective Address F4 = C F5 = 0 Rb = 0	LEA	Loads the A register with the effective address. Note that r (relative), i (indirect), and Ra (index register) can not all be zero. The contents of storage or any other registers are not altered.
Load Register F4 = C F5 = 0 Rb = 1, 2, 3, 4, 5, 6, or 7 r = 1, 2, 3, 4, Q, A, or I	LRr	Load register r with the contents of the storage location specified by the effective address. The contents of storage are not altered.
Store Register F4 = C F5 = 1 Rb = 1, 2, 3, 4, 5, 6, or 7 r = 1, 2, 3, 4, Q, A, or I	SRr	Store the contents of register r in the storage location specified by the effective address. The contents of register r are not altered.
Load Character to A F4 = C F5 = 2	LCA	Load bits A00 through A07 with a character from the storage location specified by the sum of the effective address and bits 1 to 15 of register Rb. Register Rb bit 0 set to 0 specifies the left character (bits 8 to 15 of the storage location; bit 0 set to 1 specifies the right character (bits 0 to 7). Bits A08 through A15 are cleared to zero. The contents of storage are not altered.
Store Character from A F4 = C F5 = 3	SCA	Store the contents of bits A00 through A07 into a character of the storage location specified by the sum of the effective address and bits 1 to 15 of register Rb. If bit 0 of register Rb is set to 0, the left character (bits 8 to 15) of the storage location is specified; if bit 0 is set to 1 the right character (bits 0 to 7) is specified. The contents of register A and other storage characters are not altered.
OR Register F4 = D F5 = 0 Rb = 1, 2, 3, 4, 5, 6, or 7 r = 1, 2, 3, 4, Q, A, or I	ORr	Form the logical sum (inclusive OR), bit-by-bit, of the contents of the storage location specified by the effective address and the contents of register r. The result replaces the contents of register r. The contents of storage are not altered.
OR Memory F4 = D F5 = 1 Rb = 1, 2, 3, 4, 5, 6, or 7 r = 1, 2, 3, 4, Q, A, or I	OMr	Form the logical sum (inclusive OR), bit-by-bit, of the contents of the storage location specified by the effective address and the contents of register r. The result replaces the contents of the storage location specified by the effective address. The original contents of the storage location (specified by the effective address) replaces the contents of register A. The contents of register r are not altered unless r is the A register. Memory is locked until completion of the instruction is useful for communication between processors via memory.
Compare Register Equal F4 = E F5 = 0 Rb = 1, 2, 3, 4, 5, 6, or 7 r = 1, 2, 3, 4, Q, A, or I	CrE	Skip one location if the contents of register r and the contents of the storage location specified by the effective address are equal, bit-by-bit. If they are not, execute the next instruction. The contents of register r and storage are not altered. CAUTION Each compare instruction assumes that a one-word instruction follows it.

TABLE 4-9. ENHANCED STORAGE REFERENCE INSTRUCTIONS (Contd)

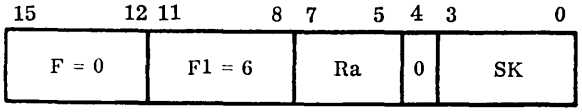
Instruction	Mnemonic	Description
Compare Register Not Equal F4 = E F5 = 1 Rb ⟶ r	CrN	If the contents of the r register and the contents of the storage location specified by the effective address are not equal, this instruction skips one location; otherwise, the next instruction is executed. The contents of the r register and storage are not altered.
Compare Character Equal F4 = E F5 = 2	CCE	Skip one location if the contents of bits 0 to 7 of register A and the character of the storage location specified by the sum of the effective address and bits 1 to 15 of register Rb are equal, bit-by-bit. If they are not, execute the next instruction. If bit 0 of register Rb is set to 0, the left character (bits 8 to 15) of the storage location is specified; if bit 0 is set to 1, the right character (bits 0 to 7) is specified. The contents of register A and storage are not altered.
Compare Character Not Equal F4 = E F5 = 3 Rb = Character flag+index	CCN	If bits 0 through 7 of the A register, the character of the storage location specified by the sum of the effective address, and bits 1 through 15 of the Rb register are not equal, the instruction skips one location; otherwise, the instruction executes the next instruction. Bit 0=0 of the Rb registers specifies the left character (bits 8 through 15) of the storage location; bit 0=1 specifies the right character (bits 0 through 7). The contents of the A register and storage are not altered.
Compare Register Greater Than or Equal F4 = F F5 = 0 Rb ⟶ r	CrG	If the contents of the r register are algebraically greater than or equal to the contents of the storage location specified by the effective address, this instruction skips one location; otherwise, the next instruction is executed. The contents of the r register and storage are not altered.
Compare Register Less Than or Equal F4 = F F5 = 1 Rb ⟶ r	CrL	If the contents of the r register are algebraically less than or equal to the contents of the storage location specified by the effective address, this instruction skips one instruction; otherwise, the next instruction is executed. The contents of the r register and storage are not altered.

- 16-bit storage (r = 0, i = 1, and Δ = 0) – The base address equals the contents of P + 2. The contents of index register Ra, when specified, are added to the base address to form the effective address.
- 16-bit relative (r = 1, i = 0, and Δ = 0) – The base address equals the contents of P + 2 plus P + 2. The contents of index register Ra, when specified, are added to the base address to form the effective address.
- 16-bit relative indirect (r = 1, i = 1, and Δ = 0) – The address of the third word of the instruction, P + 2, plus the contents of the third word of the instruction is an indirect address. The content of this address is the base address. The contents of the index register Ra, when specified, are added to the base address to form the effective address.

The instruction descriptions are given in table 4-9.

DECREMENT AND REPEAT

These instructions have the following format:



Decrement and repeat instructions are specified when the F field is 0, the F1 field is 6, bit 4 is 0, and the Ra and SK fields are not both 0. (If these fields are both 0, the instruction is an SPB.)

Decrement and repeat instructions contain two parts: register field (ra) and skip count (SK). The register field specifies which register is to be decremented by one and checked for the skip condition. The skip count specifies how many locations to repeat (go backwards) if the skip condition is met.

When the skip condition is met, the skip count is subtracted from the P register to obtain the address of the next instruction (for example, when the skip count is one, go to P - 1). When the skip condition is not met, the address of the next instruction is P + 1. The skip count does not have a sign bit.

Note that Ra = 0 and bit 4 = 1 are reserved for future expansion.

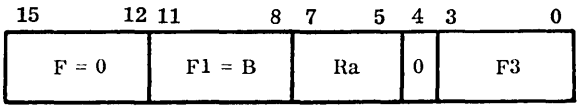
The instruction description is:

Decrement and Repeat if Positive DrP SK
Ra = 1, 2, 3, 4, 5, 6, or 7
r = 1, 2, 3, 4, Q, A, or I

Decrement the contents of register r by one. Operation on overflow is the same as for the ADD instruction. Repeat (go backwards) SK locations if the contents of register r are positive (bit 15 is 0); otherwise, execute the next instruction.

MISCELLANEOUS INSTRUCTIONS

Miscellaneous instructions have the following format:



Miscellaneous instructions are specified when the F field is 0, the F1 field is equal to decimal 11 (hexadecimal B), bit 4 is 0, and the Ra and F3 fields are not both 0. (If these fields are both 0, the instruction is an NOP.) All of the miscellaneous instructions are privileged instructions; that is, if they are executed by an unprotected program, they cause a program protect violation.

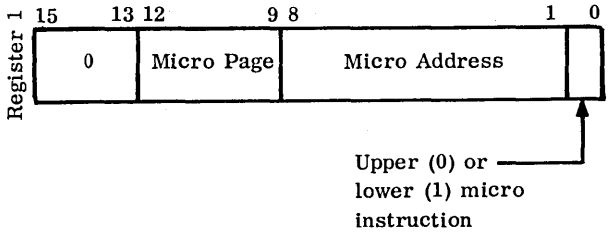
Miscellaneous instructions contain two parts: operation field (F3) and register field (Ra).

If Ra is nonzero, the F3 operation field can select up to 16 miscellaneous instructions with register Ra used to specify an operand. If Ra is 0, the F3 operation field can select up to 15 more miscellaneous instructions without any explicit operand specified.

All the miscellaneous instruction descriptions are given in table 4-12. Those instructions that require more detail are described below. Instructions LRG and SRG are shown in figures 4-1 and 4-2, respectively.

The miscellaneous instructions formats are:

1. Load Micro Memory



Initially, the Q register contains the number of 32-bit micro-memory instructions to be transferred (if Q = 0, no instructions are transferred). Register 1 contains the starting address of micro memory. Register 2 contains the starting address of processor main memory.

The most significant bit (15) of the contents of the starting address is transferred to the most significant bit of the first micro instruction. The least significant bit (0) of the contents of the starting address plus one is transferred to the least significant bit. This instruction is interruptible after storing each 32-bit micro memory instruction and when registers 1, 2, and Q are incremented/decremented to allow the instruction to be restarted after any interruption. When the instruction is completed, these registers contain the following rather than their original values:

Where: i is the initial value before execution.

2. Set/Sample Output or Input

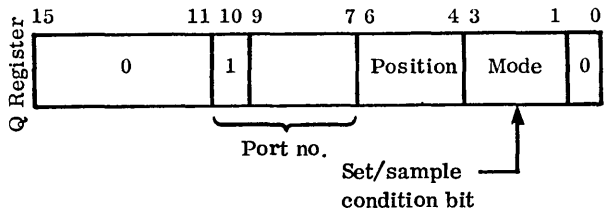


TABLE 4-12. MISCELLANEOUS ENHANCED INSTRUCTIONS

Instruction	Mnemonic	Description
Load Micro Memory F3 = 1 Ra = 0	LMM	Load a 32-bit micro-memory instruction into read/write micro memory from 16-bit processor main memory. (For read-only micro memory or no micro memory, no operation is executed.)
Load Registers F3 = 2 Ra = 0	LRG	Registers 1, 2, 3, 4, Q, A, I, M, and the overflow indicator are loaded with the contents of nine storage locations, beginning at a storage location specified by the contents of the contents of the next location, P + 1. The contents of the nine storage locations will not be altered and the next instruction will be executed at location P + 2 (i.e., the LRG instruction is a two-word instruction). Refer to figure 4-1.
Store Registers F3 = 3 Ra = 0	SRG	Registers 1, 2, 3, 4, Q, A, I, M, and the overflow indicator are stored into nine storage locations specified by the contents of the next location, P + 1, incremented by a decimal 10. The contents of the registers will not be altered and the next instruction will be executed at location P + 2 (i.e., the SRG instruction is a two-word instruction). Refer to figure 4-2.
Set/Sample Output or Input F3 = 4 Ra = 0	SIO	Set one word from register A for output to an external device. The word in register Q selects the receiving device. For input, one word from an external device is sampled (input) to register A. The word in register Q selects the sending device.
Sample Position/Status F3 = 5 Ra = 0	SPS	Sample (input) to the A register the position and status of a M05 device, which has caused a processor macro interrupt. The word in the Q register selects the device. The instruction also provides for clearing the M05-generated processor macro interrupt.
Define Micro Interrupt F3 = 6 Ra = 0	DMI	Define the use of one of the 12 available micro interrupts. (The use of micro interrupts 12 through 15 is restricted for internal use.) This instruction allows a micro interrupt to be enabled/disabled and defined for auto-data transfer (ADT) or special usage.
Clear Breakpoint Interrupt F3 = 7 Ra = 0	CBP	Clear the processor macro breakpoint interrupt. This interrupt occurs when the following conditions are true: macro breakpoint is externally selected, macro breakpoint interrupt option is externally selected, the MP recognizes a breakpoint condition and generates a macro breakpoint interrupt because of b.
Generate Character Parity Even F3 = 9 Ra = 0	GPE	Set or clear bit 7 of the A register so that bits 0 to 7 have an even parity. The other bits in the A register are not altered.
Generate Character Parity Odd F3 = 8 Ra = 0	GPO	Set or clear bit 7 of the A register so that bits 0 through 7 have an even parity. The other bits in the A register are not altered.
Scale Accumulator F3 = A Ra = 0	ASC	Shift the A register left (end-around) until bits 14 and 15 of the A register are different. Upon completion of the instruction, register 1 contains the number of spaces that the A register was shifted. (This number may range from 0 to 14.) If the A register is +0 (000 or $FFFF_{16}$), no shift has been performed and register 1 contains -0 ($FFFF_{16}$).

TABLE 4-12. MISCELLANEOUS ENHANCED INSTRUCTIONS (Contd)

Instruction	Mnemonic	Description
Absolute Page Mode F3 = B Ra = 0	APM	Absolute page mode is specified. Only the first 65,536 words of memory can be addressed.
Page Mode 0 F3 = C Ra = 0	PM0	Page mode 0 is specified. Segments of 2,048 words, totaling 65,536 words can be addressed via the 32-page mode 0 register file.
Page Mode 1 F3 = D Ra = 0	PM1	Page mode 1 is specified. Segments of 2,048 words, totaling 65,536 words can be addressed via the 32-page mode 1 register file.
Load Status/Mode Registers F3 = E Ra = 0	LSM	Status/mode register 1 is loaded into the A register; status/mode register 2 is loaded into the Q register.
Load Upper Unprotected Bounds F3 = 0 Ra⟶R	LUB R	Load the upper unprotected bounds register from the contents of the R register.
Load Lower Unprotected Bounds F3 = 1 Ra⟶R	LLB R	Load the lower unprotected bounds register from the contents of the R register.
Execute Micro Sequence F3 = 2 Ra⟶R	EMS R	Transfer machine control to the upper micro instruction of the page/micro-memory address in bits 0 through 15 of register R. A section of micro memory is assumed to have been previously loaded.
Write Page Register F3 = 3 Ra⟶R	WPR R	Write the contents of the R register, bits 0 through 8, into the page register specified by bits 10 through 15 of the R register. Bit 10 is zero or one for a page mode 0 or 1 register respectively. Bits 11 through 15 specify one of the 32-page registers within the page mode. Bit 9 is unused and should be zero.
Read Page Register F3 = 4 Ra⟶R	RPR R	Read the contents of the page register specified by bits 10 through 15 of the R register into the A register. Bit 10 is zero or one for a page mode 0 or 1 register respectively. Bits 11 through 15 specify one of the 32 page registers within the page mode. Bits 0 through 9 are unused and should be zero.
Read ECC Status F3 = 5 Ra⟶R	ECC R	Read the error correction code (ECC) status of the memory word, specified by the address in the R register, into the A register.
Write Page File F3 = 6 Ra⟶R	WPF R	Write N page registers, where N is the contents of the Q register. The data to be written is contained in a table whose address is in the R register.
Read Micro-Memory Data F3 = 7 Ra⟶R	RMD R	Sixteen bits of micro-memory data are read and loaded into the A register. The R register contains the micro-memory address.
Set Status/Mode 1 F3 = 8 Ra⟶R	SS1 R	Forms the logical sum (inclusive OR), bit-by-bit, of the R register and status/mode register 1 (SM1). The result replaces the original contents of SM1. The R register is not altered.

TABLE 4-12. MISCELLANEOUS ENHANCED INSTRUCTIONS (Contd)

Instruction	Mnemonic	Description
Set Status/Mode 2 F3 = 9 Ra⟶R	SS2 R	Forms the logical sum (inclusive OR), bit-by-bit, of the R register and status/mode register 1 (SM2). The result replaces the original contents of SM2. The R register is not altered.
Clear Status/Mode 1 F3 = A Ra⟶R	CS1 R	Forms the logical product (AND), bit-by-bit, of the complement of the R register and status/mode register 1. The result replaces the original contents of SM1. The R register is not altered.
Clear Status/Mode 2 F3 = B Ra⟶R	CS2 R	Forms the logical product (AND), bit-by-bit, of the complement of the R register and status/mode register 2. The result replaces the original contents of SM2. The R register is not altered.

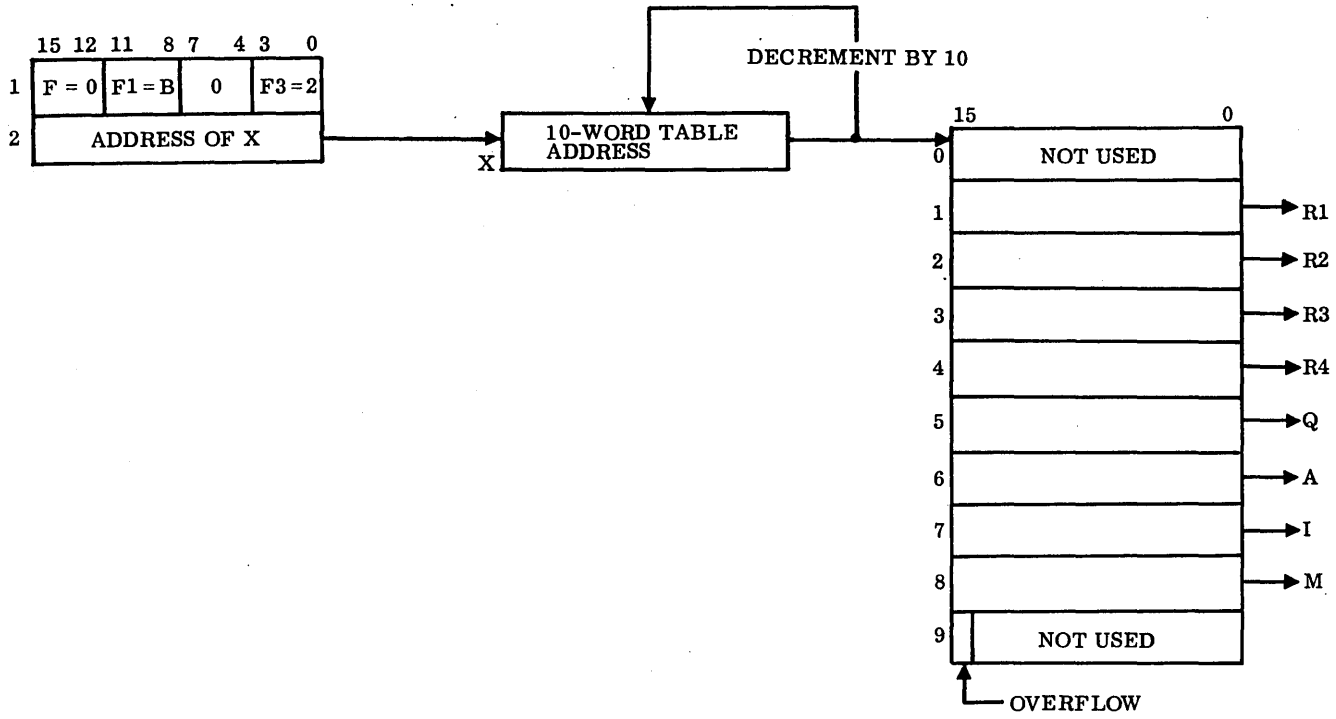


Figure 4-1. LRG Instruction

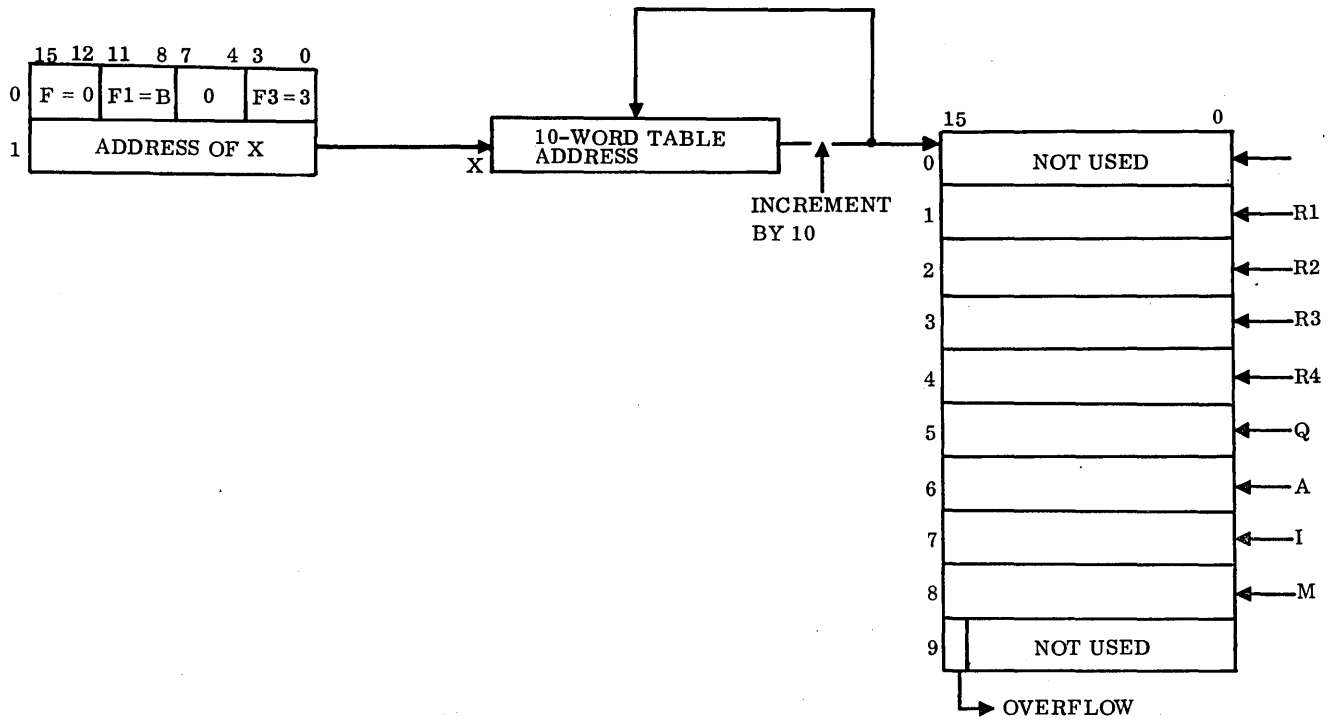
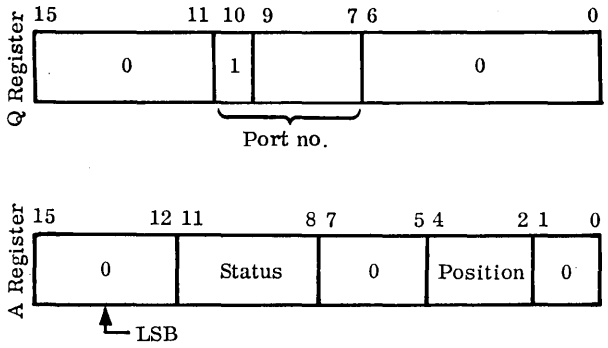
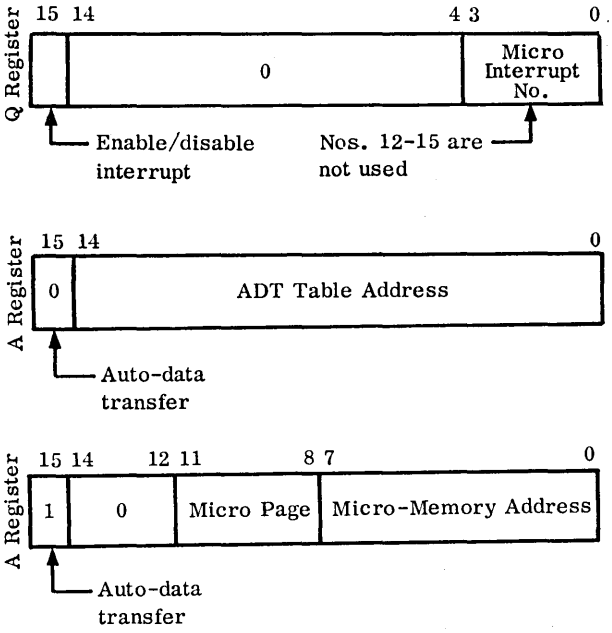


Figure 4-2. SRG Instruction

3. Sample Positions Status



4. Define Micro Interrupt

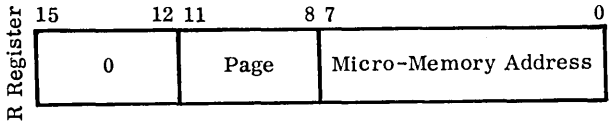


When bit 15 of the A register is set to a 1, a jump is made to the upper micro instruction of the page/micro memory in bits 0 to 14. A section of micro memory is assumed to have been previously loaded, and it must process the micro interrupt properly and return control to the current macro instruction address (P) by jumping to the lower micro instruction of micro-memory address $3E_{16}$ in micro page 0. Registers P, A, Q, and all of file 2 should not be altered, and return must be within 12.5 microseconds.

CAUTION

The processor micro function, SUB-, must not be used. Extreme caution should be exercised in using this option, since it provides an escape from the 1700 emulation being performed.

5. Execute Micro Sequence



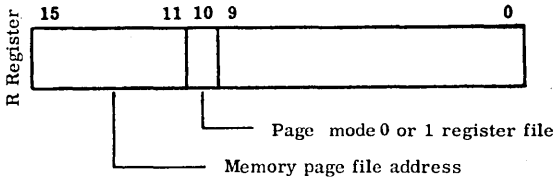
An execute micro sequence to nonexistent micro memory may cause an indeterminate result. Control should be returned to the next macro-instruction address (P + 1) by jumping to the lower micro instruction of micro-memory address $3E_{16}$ in micro page 0. Registers P, A, Q, and all of file 2 should not be altered, and return must be within 12.5 microseconds (or the micro sequence must be interruptable).

Bits 12 through 15 must be zero. Bits 8 through 11 specify the micro page. Bits 0 through 7 specify the micro-memory address.

CAUTION

Extreme caution should be exercised in utilizing this option, since it provides an escape from CYBER 18/1700 emulation.

6. Write Page Register

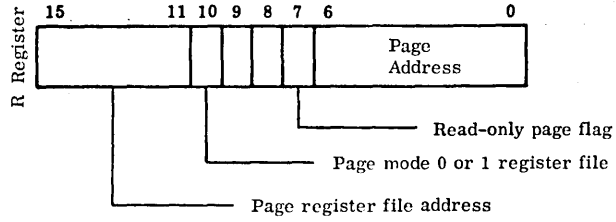


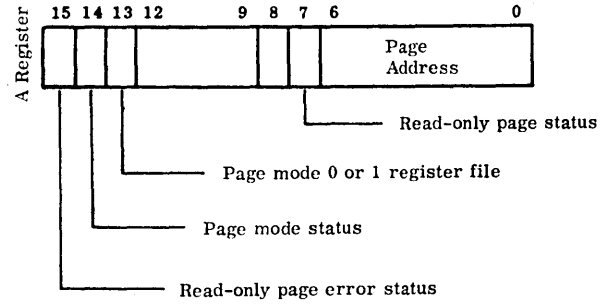
This command loads a page address into a page register file (memory page file) in main memory.

When bit 10 of the R register is zero, one of the 32 locations that corresponds to the page mode 0 register file specified by bits 11 through 15 is loaded with the information contained in bits 0 through 8 of the R register. If bit 10 is a one, one of the 32 locations that corresponds to the page mode 1 register file is loaded. Bits 0 through 8 represent the page address, which forms the seven most significant bits of the 18-bit main memory address (256K-word addressing). Bit 7, if a one, establishes the page as a read-only page (see section 6, Program Protect). To write into any page location on a page mode main memory write reference, this bit must be set to 0. Bit 9 is unused and should be zero.

A write page register instruction clears a read-only page interrupt.

7. Read Page Register



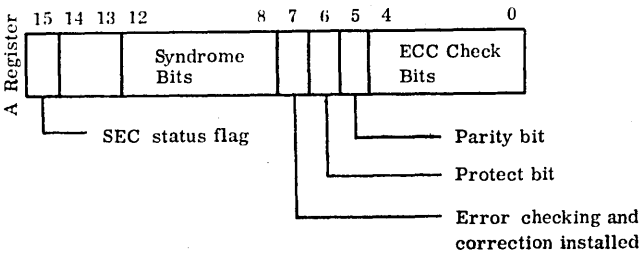


The page register is referenced as described for a write page register instruction.

Bits 0 through 8 of the A register are loaded with the contents of the referenced page register. Bits 0 through 6 contain the 7-bit stored page address; bit 7 contains the read-only page bit for that page. Bits 8 through 12 are unused and are zero.

Bits 13 through 15 of the A register contain status information from main memory. Bit 13 indicates the page mode register set selected. Bit 14 shows that main memory is in absolute mode if zero, page mode if one. Bit 15 indicates a read-only page error. A read page register instruction clears this error bit.

8. Read ECC Status



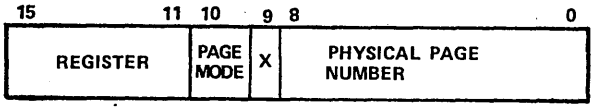
This instruction loads main memory status information into the A register. The address in register R may be either absolute or page mode format, appropriate for the main memory mode previously selected.

In the A register, bits 0 through 4 are the five ECC bits for the addressed word. Bit 5 is the parity bit, and bit 6 is the program protect bit. Bit 7 is a one if the ECC option is installed.

Bits 8 through 12 are the ECC syndrome bits, which should all be zero if no memory errors exist. Bit 15 is the single error correction (SEC) status flag, which is set if a single bit main memory error has occurred on a read or normal write operation. The single error correction status flag is set on single bit errors only if ECC is installed (the error is automatically corrected). If ECC is not installed, a single bit error causes a parity error.

A Read ECC Status instruction clears the single error correction status flag in main memory.

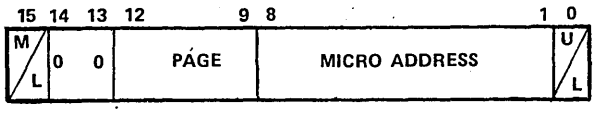
9. Write Page File



Each datum has the following format: The physical page number in bits 0 through 8 is written into the page register specified by bits 10 through 15. Bit 10 is zero or one for a page mode 0 or 1 register respectively. Bits 11 through 15 specify one of the 32-page registers within the page mode. Bit 9 is unused and should be zero.

The data need not be ordered by page register. If the Q register is zero, the instruction is ignored.

10. Read Micro-Memory Data



Bit 15 specifies the most (0) or least (1) significant 16 bits of the micro instruction. Bits 13 and 14 must be zero. Bits 9 through 12 specify the micro page. Bits 1 through 8 specify thge micro-memory address. Bits 0 specifies the upper (0) or lower (1) micro instruction.

AUTO-DATA TRANSFER

Auto-data transfer provides for pseudo direct memory transfers of data blocks to or from a device. At the macro level, the transfer appears as a direct memory access (DMA) transfer. At the micro level, the 1700 emulator

processes each data interrupt and inputs or outputs the next data in a singular fashion. Thus, auto-data transfer takes less time than input/output via the INP, OUT, or SIO instructions, but more time than a true DMA transfer.

To accomplish an auto-data transfer for a particular device, perform the following:

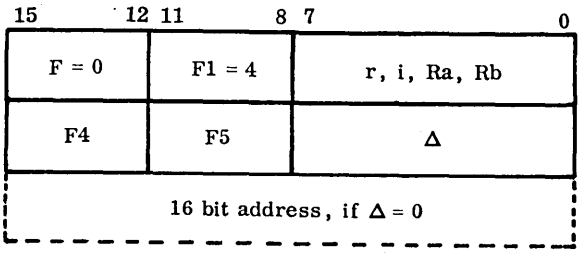
- The device and its controller must be capable of operating in the auto-data transfer mode.
- The macro programmer must initialize the auto-data transfer that is defined by the DMI instruction.
- The macro programmer must execute a DMI instruction. This command specifies where the block of data is, how long it is, the direction (input/output), and the device's address.
- The auto-data transfer operation is then initiated by an INP, OUT, or SIO instruction as specified by the particular device.

While the auto-data transfer operation is in progress, the emulator is executing instructions. After each instruction is executed, interrupts are checked. When the particular auto-data transfer micro interrupt becomes the highest active interrupt, the next data is input or output. After the interruption, the next instruction is executed, except when another interrupt is active.

When the auto-data transfer operation is completed (or if there is an error), a macro interrupt is generated. The macro programmer may then disable the auto-data transfer micro interrupt or initiate another auto-data transfer operation to or from the device. For MOS devices, an SPS instruction must be performed to clear the macro interrupt.

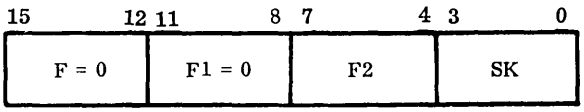
ENHANCED INSTRUCTIONS

ENHANCED STORAGE REFERENCE (r,i, Ra, Rb=0)



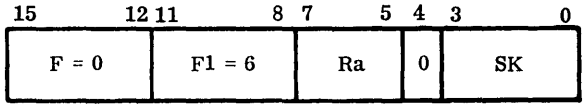
F4 = 5, F5 = 0, Rb = 0	SJE
F4 = 5, F5 = 0, Rb = 0	SJr
F4 = 8, F5 = 0, Rb = 0	ARr
F4 = 8, F5 = 1, Rb→r	RMr
F4 = 9, F5 = 0, Rb = 0	SBr
F4 = 9, F5 = 1, Rb→r	SMr
F4 = A, F5 = 0, Rb = 0	ANr
F4 = A, F5 = 1, Rb = 0	AMr
F4 = B, F5 = 0, Rb→r	ERr
F4 = B, F5 = 1, Rb→r	EMr
F4 = C, F5 = 0, Rb = 0	LEA
F4 = C, F5 = 0, Rb = 0	LRr
F4 = C, F5 = 1, Rb = 0	SRr
F4 = C, F5 = 2	LCA
F4 = C, F5 = 3	SCA
F4 = D, F5 = 0, Rb = 0	ORr
F4 = D, F5 = 1, Rb = 0	OMr
F4 = E, F5 = 0, Rb = 0	CrE
F4 = E, F5 = 1, Rb→r	CrN
F4 = E, F5 = 2	CCE
F4 = E, F5 = 3, Rb =→ r	character flag + index CCN
F4 = F, F5 = 0, Rb→r	CrG
F4 = F, F5 = 1, Rb→r	CrL

ENHANCED SKIP INSTRUCTIONS



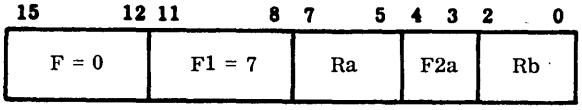
F2 = 0, 4, 8, or C F4 = 4, 1, 2, or 3	SrZ SK
F2 = 1, 5, 9, or D r = 4, 1, 2, or 3	SrN SK
F2 = 2, 6, A, or E r = 4, 1, 2, or 3	SrP SK
F2 = 3, 7, B or F r = 4, 1, 2, or 3	SrM SK

DECREMENT AND REPEAT



Ra = 1 to 7 DrP SK

ENHANCED INTER-REGISTER

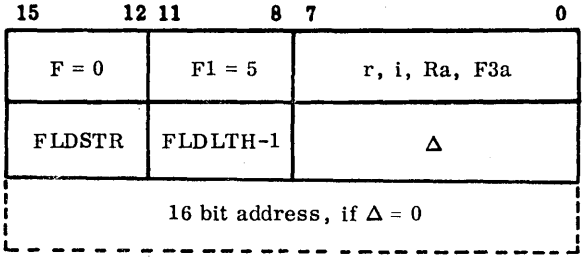


F2a = 0, Ra = 1-7 XFr R
Rb = 1-4, Q, A, I

NOTE

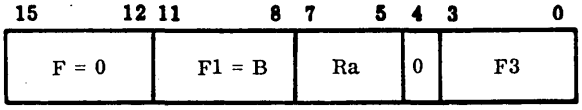
Ra, Rb (0 to 7) = 0, 1, 2, 3, 4, Q, A, I

FIELD REFERENCE



F3a = 2	SFZ
F3a = 3	SFN
F3a = 4	LFA
F3a = 5	SFA
F3a = 6	CLF
F3a = 7	SEF

MISCELLANEOUS



F3 = 1, Ra = 0	LMM
F3 = 2, Ra = 0	LRG
F3 = 3, Ra = 0	SRG
F3 = 4, Ra = 0	SIO
F3 = 5, Ra = 0	SPS
F3 = 6, Ra = 0	DMI
F3 = 7, Ra = 0	CBP
F3 = 8, Ra = 0	GPE
F3 = 9, Ra = 0	GPO
F3 = A, Ra = 0	ASC
F3 = B, Ra = 0	APM
F3 = C, Ra = 0	PM0
F3 = D, Ra = 0	PM1
F3 = E, Ra = 0	LSM
F3 = 0, Ra→R	LUB R
F3 = 1, Ra→R	LLB R
F3 = 2, Ra→R	EMS R
F3 = 3, Ra→R	WPR R
F3 = 4, Ra→R	RPR R
F3 = 5, Ra→R	ECC R
F3 = 6, Ra→R	WPF R
F3 = 7, Ra→R	RMD R
F3 = 8, Ra→R	SS1 R
F3 = 9, Ra →R	SS2 R
F3 = A, Ra→R	CS1 R
F3 = B, Ra→R	CS2 R

COMMENT SHEET

MANUAL TITLE CDC® CYBER 18 Processor with MOS Memory Macro Level

Hardware Reference Manaul

PUBLICATION NO. 96768300 REVISION D

FROM NAME: _____

BUSINESS
ADDRESS: _____

COMMENTS: This form is not intended to be used as an order blank. Your evaluation of this manual will be welcomed by Control Data Corporation. Any errors, suggested additions or deletions, or general comments may be made below. Please include page number.

☐ Please Reply ☐ No Reply Necessary

CUT ALONG LINE

STAPLE

STAPLE

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 8241 MINNEAPOLIS, MINN.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
PUBLICATIONS AND GRAPHICS DIVISION
4455 EASTGATE MALL
LA JOLLA, CALIFORNIA 92037

CUT ALONG LINE

FOLD

FOLD